

Searching for MobileNetV3

Andrew Howard¹ Mark Sandler¹ Grace Chu¹ Liang-Chieh Chen¹ Bo Chen¹ Mingxing Tan²
Weijun Wang¹ Yukun Zhu¹ Ruoming Pang² Vijay Vasudevan² Quoc V. Le² Hartwig Adam¹

¹Google AI, ²Google Brain

{howarda, sandler, cxy, lcchen, bochen, tanmingxing, weijunw, yukun, rpang, vrv, qvl, hadam}@google.com

ICCV 2019

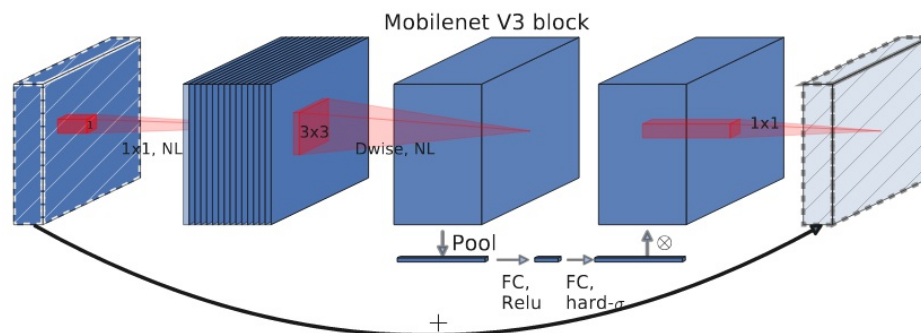
目的：

在 mobileNetV2 的基础上，引入了 SE 通道注意力结构，结合网络架构搜索；并修改了网络尾部和头部结构，降低计算量；用分段函数近似 swish 激活函数。

方法：

mobileNetV3 的基本模块：

使用深度可分离卷积，在第一个 1*1 卷积将特征扩展到高维，对每个通道使用同个卷积，第二个 1*1 卷积将特征缩小到原来大小后使用线性激活函数，因为 relu 对低维特征造成容易特征损失。SE 模块加在深度可分离卷积之后，对不同通道的特征进行增强。

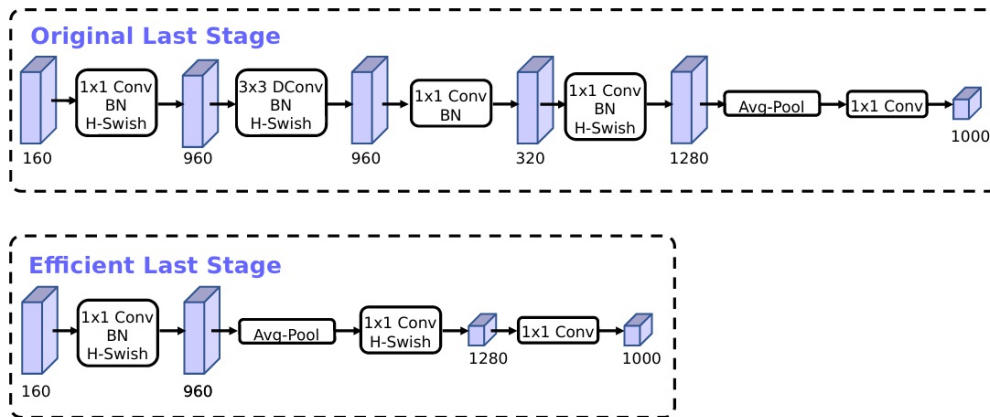


网络架构搜索：

资源受限的 NAS：计算和参数量受限的前提下搜索网络的各个模块，所以称之为模块级的搜索。NetAdapt：用于对各个模块确定之后网络层的微调。（这部分不熟）

修改尾部结构：

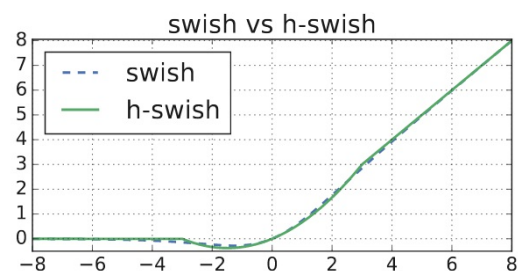
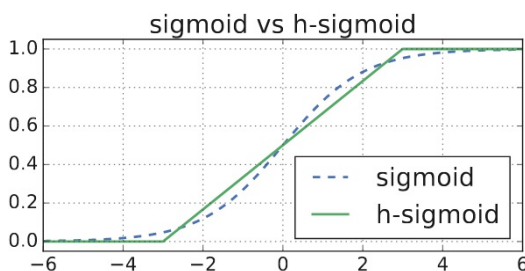
首先利用 avg pooling 将特征图大小由 7x7 降到了 1x1，降到 1x1 后，然后再利用 1x1 提高维度，这样就减少了 $7 \times 7 = 49$ 倍的计算量。并且为了进一步的降低计算量，直接去掉了前面纺锤型卷积的 3x3 以及 1x1 卷积。



近似非线性激活函数：

swish 激活函数能够提高网络的精度，但是在移动端就比较耗时；文中用 relu6 近似 sigmoid 得到 h-swish，它能提高计算效率，同时消除了潜在的精度损失。

$$\text{h-swish}[x] = x \frac{\text{ReLU6}(x + 3)}{6}$$



总结：

文章的重点是在神经网络的架构搜索，硬件依赖太大估计是用不了的。其中的 swish 激活函数和 SE 结构可以尝试用一下，毕竟不会增加太大的计算量。